Apple II

Technical Bulletin

August–September 1990 Issue 11

Feature Articles

Tech Sheets

Networking and Connectivity

Power User Tips

Workarounds

Special Announcements

Annual Subscription: $129

Demystifying the GS/OS Cache

The following article was written by Matt Deatherage for develop, The Apple Technical Journal, *and is reprinted here with permission. For more information about* develop, *see the note at the end of the article.*

GS/OS® has given Apple IIGS® users an important capability: caching. To the newly initiated, the mysteries of the GS/OS cache may still seem profound. This article clarifies the basics of the caching algorithm and offers useful pointers for working safely and efficiently.

In the past couple of years, the Apple IIGS system software has grown by leaps and bounds. With the introduction of thousands of new features and capabilities, one very important one often goes unnoticed. GS/OS is the first Apple® II operating system of any kind to provide a comprehensive caching implementation.

Although the cache is an important part of GS/OS, its purpose and nature are sometimes misunderstood. Some people create huge caches in the hope that peripherals will start behaving like RAM disks. Others set the cache size to zero because they think it's wasting time and memory. Some developers mistakenly believe their program has absolute control over caching, which is true only if they're writing device drivers. This article reaches into the murky depths of your IIGS® computer's memory and allows you to examine the cache in the light you would normally use, say, to read *develop* by.

Cache Fundamentals

Let's start at the beginning, with the basics. What's a cache? How is it managed? How big is it? How does the GS/OS cache differ from that of the Macintosh? Read on.

What's a Cache?

A cache is a part of memory in which the operating system can keep a spare copy of information read from and written to a device, in an attempt to decrease disk access time.

Many people use RAM disks to decrease disk access time. Because the operating system doesn't actually have to physically manipulate any media to access information on a RAM disk, the information is saved and retrieved at a very nice clip, exceeded only by the speed at which the operating system can read information already in main memory. In a caching algorithm, that is exactly what happens.

When information is read from a device, the operating system reads from the physical media and returns the information to the caller who asked for it. Additionally, the operating system keeps a spare copy of the information in the cache. The next time a caller requests information, the operating system first looks in the cache to see if it already has it from a previous media access. If it's there, the operating system simply moves the data into the caller's buffer, completing the call without having to actually read anything from disk.

Information to be written can also be cached. The operating system writes the information to the disk and also writes it to the cache. This method, called a write-through cache, ensures that the information in the cache is never more recent than the information on the disk. GS/OS uses a write-through cache except in one special situation—during a write-deferral session.

The Cache Manager and What It Caches

In GS/OS, several distinct managers handle different parts of the operating system or environment. Many of these managers (the Loader and the Device Manager, for example) are familiar to application and driver authors. Another, more obscure manager takes care of the cache; plainly enough, it's called the Cache Manager. The Cache Manager handles the storage of the cache and all requests involving it.

The GS/OS Cache Manager caches only one thing: blocks. Blocks are what both generated and loaded drivers read from block devices. They can be traditional ProDOS®-sized 512-byte blocks from traditional ProDOS devices; they can be 512-byte blocks from nontraditional ProDOS devices, such as network volumes or nondisk devices; or they can be odd-sized blocks from odd peripherals. If you had a GS/OS loaded driver to read 981-byte blocks from a 20-gigahertz, 35-terabyte RAM disk hooked to your Apple IIGS through a slot-based card, GS/OS could cache those blocks. (Whether or not any file system translators in the system could use the device is another story.) The Cache Manager does not cache anything having to do with printers, modems, or other character devices.

The Size of the Cache

The size of the cache is determined by a battery RAM (BRAM) parameter. When the Cache Manager is initialized as part of the GS/OS boot process, it retrieves this value from the battery backed-up RAM and adjusts so that the given value is taken as the maximum size to which the cache can grow. The BRAM parameter is currently number $81 and represents the cache size in 32K increments (a value of 2 would indicate 64K). You should know that the location and interpretation of the cache size parameter are not guaranteed at this point, and relying on the location or interpretation of the parameter might get you into trouble. When they are guaranteed, Apple II Developer Technical Support will release a Technical Note detailing this point, or the *GS/OS Reference* will be revised to document it.

The size of the cache is not an exact number but an exact maximum, and it is that only when nonzero. If the user has set the cache size to nonzero, that value (which will always be an increment of 32K) is the maximum cache size. If the user has set the cache size to zero, GS/OS allocates a 16K cache for system purposes—so that system components such as file system translators (FSTs) and drivers (both generated and loaded) can take advantage of the speed increase the cache provides. There is no way to completely turn off caching under GS/OS.

When the cache is initialized, it is empty and has zero size. As blocks are added to it, it grows as necessary to accommodate the increased use until the maximum size is reached. If no one ever asks for blocks to be cached, the cache remains empty and occupies no memory.

What Happens When the Cache Is Full

Having a maximum cache size implies that there can be a problem—what happens when nothing else can be written to the cache because there is no room in cache memory for more information? How does GS/OS behave when there's no room in the Cache Inn?

The operating system, through the cache algorithm, has a few options for handling the possibility that the cache will be full. The most obvious option is for the algorithm to shrug its bit-encoded shoulders and say, "Well, the cache is full, so nothing else will be written to it." Information that comes knocking at the door thereafter is written to disk only and read from disk each time it's needed.

Most caching algorithms, including the one used by GS/OS, are a little bolder. They attempt to identify which blocks in the cache have actually been used and to keep those blocks in the cache, simultaneously removing from the cache the blocks that have not been used. The rationale for this is that if a block hasn't been read from the cache in a long time, it's probable that no one's going to want to read it again for a while, and the cache can be better used by a new block. For example, when GS/OS is booted, the file START.GS.OS is read from the disk. Suppose, for the sake of argument, that this file's information was placed in the cache. This file isn't likely to be read again by the operating system or by any application unless GS/OS has to be reloaded, which under version 5.0 and later never happens. This file could, therefore, be sitting in the cache, taking up valuable cache space that subsequently can't be used by directory blocks, bitmap blocks, or even program files that will be needed repeatedly, such as the Finder™or APW™commands. The strategy used by most caching algorithms tries to keep space in the cache free for use by directory blocks, bitmap blocks, and program files that will be needed repeatedly.

One popular caching algorithm keeps all the blocks in a chain, with the most recently used block at the beginning of the chain and the block not used for the longest time at the end. When there's no more room in the cache, the blocks at the end of the chain are removed, making room for new blocks. This algorithm, straightforwardly called a least recently used (LRU) caching algorithm, is what GS/OS uses.

Some blocks, however, can't be kicked out of the cache by the LRU algorithm. These are the blocks placed in the cache by a write-deferral session. When such a session is in progress, some or all of the information written to files is write-deferred, which means GS/OS keeps the information around in the cache instead of taking the time to write it all to physical media. This is accomplished in different ways for different file systems; each file system translator behaves in the way it can achieve the best performance during a session. For example, the ProDOS FST writes actual data to the disk but not system-level information such as directories or bitmap blocks.

If any write-deferred blocks are in the cache, the Cache Manager will not purge them to make room for new blocks until they have been written to disk. If the cache fills up with nonpurgeable blocks and another nonpurgeable block must be added, a feature known as AutoFlush, which was new to System Software 5.0, takes over, stops the session, and flushes all write-deferred blocks to disk so that new ones can be added. This has the effect of breaking a session into lots of mini-sessions, each exactly long enough to get the best possible use from the cache.

Blocks in the cache can also be deleted from it by another method besides the LRU algorithm. If a driver detects a disk-switched condition, which normally means that an on-line volume has been taken off-line, it makes the `SET_DISKSW` System Service call, which enables the file system translators to remove all blocks belonging to the switched disk from the cache so that no blocks are in the cache for volumes that aren't currently on-line. This ensures that no one accidentally reads from the device and gets an old block from the cache, or that no blocks are in the cache for volumes that aren't currently online.

Differences from the Macintosh

The GS/OS Cache Manager is different from the caching implementation on the Macintosh® in two important ways. The first difference is in the way memory is allocated. On the Macintosh, if the user sets the disk cache size to 128K, a 128K block of memory stays allocated for the disk cache unless someone resizes the cache. Under GS/OS, memory for cached blocks is allocated as needed up to the maximum set by the user. The other difference between GS/OS and Macintosh is in the size of the blocks that can be cached. The GS/OS cache can handle a block of any size; if a device deals in 2048-byte blocks—as some CD-ROM discs do—GS/OS will cache a 2048-byte block. The Macintosh Cache Manager, on the other hand, can cache only 512-byte blocks.

How Applications Request to Use the Cache

Applications—or desk accessories, inits, or anyone who makes GS/OS system calls—request to use the cache through cache-related parameters to the system calls. Specifically, applications can ask that data read from files be cached by using the `cachePriority` field of class one (and only class one) `Read` and `Write` calls.

Caching Can Be Requested by Class One Calls

An application requests caching by setting the `cachePriority` field in the GS/OS parameter block of `Read` and `Write` calls. (These are the only two GS/OS system calls with this field in the parameter block. The system calls `DRead` and `DWrite` do not have a `cachePriority` field, as the Device Manager always disables caching of blocks read or written through these calls.)

A value of $0000 for the word-length parameter in the `cachePriority` field is the norm (and the default if this parameter is omitted) and indicates that blocks involved in this call should not be cached. A value of $0001 identifies the blocks involved as candidates for caching. Only a value of $0001 in this field will cause files read at the application level to be considered for caching.

The fact that a caching request has been made by an application doesn't mean that it will be fulfilled. Applications do not call the Cache Manager; other system components do. Those components (file system translators and drivers) may deny the request when it doesn't make sense or is dangerous for the file's blocks to be cached.

Caching Cannot Be Requested by Class Zero Calls

Applications written using class zero calls (including older ProDOS 16 applications) cannot request that their files be placed in the cache. To make such a request, the application must be changed to use class one calls. At first this seems a little harsh and arbitrary, but it's quite the opposite.

GS/OS could treat class zero file calls one of two ways—it could automatically cache everything, or automatically cache nothing. If it cached everything, reading any file larger than the maximum cache size would flush all the cached blocks, no matter how frequently they were used. This would force them all to be reread from disk the next time they were needed. Such a method is normally a grand waste of time, since most files on the Apple IIGS are typically read once. Most programs still follow the old Apple II method of "read the file, modify it, and write it back," and writing such files to the cache serves only to slow things down while the flushed blocks are reread from disk. Better methods for file manipulation exist these days, but the system was designed for maximum performance using the methods of the time.

An application knows best which files it will be reading from disk often enough to benefit from caching. Such decisions are often reached after long performance studies of cached reads vs. purgeable handles in memory. For more pointers on application-level caching, see GS/OS Technical Note #3, Pointers on Caching.

Write Deferral Sessions and the Cache

Applications can also speed up disk-intensive operations through the use of write-deferral sessions. An application begins such a session by giving the GS/OS system call `BeginSession`. This places all subsequent writes in a mode where some or all of the blocks written to disk are placed only in the cache and not on the media.

Deferred blocks are then written to the media when the `EndSession` call is made. If the `EndSession` call is not made or if it's made before the files being written to are closed, some of the blocks for the files written may be on the disk while others are in the cache only. This damages disks in most file systems as fast as fingerprints on the media. Be sure *always* to issue `EndSession` calls on every exit path after a `BeginSession` call to prevent blocks written from being only in the cache and not on the disk. And be sure to close all open files before calling `EndSession`, since the operating system can get caught with write-deferred blocks in open files if the session is ended and the disk is ejected before the files are closed.

How Cache Requests Are Fulfilled

GS/OS attempts to make sure that although an application can cache blocks if it so chooses, by default the most intelligent use possible of the caching capabilities will be made. It does this by filtering out possibly bogus requests for caching at several levels. When an application makes a request to use the cache, the request is filtered through drivers and/or file system translators. These agents of the Cache Manager evaluate requests to make sure that the most intelligent possible use of the caching capability is made. If an application's request to use the cache is found to be valid, one of these system components calls a System Service routine to add blocks to the cache.

The System Service Calls

The ultimate caching authority is at the System Service call level. System Service calls are used by drivers and file system translators to access the routines that act on cache requests. These calls are available only to drivers and file system translators; they are not available to applications. System Service calls are accessed through vectors in the $01/FC00 page and are described in the *GS/OS Reference,* Volume 2, Chapter 12.

The following four System Service calls—to add blocks, find blocks in the cache, move blocks in and out of the cache, and remove all purgeable blocks in the cache belonging to a switched disk—are the only calls that can be made by drivers. Other cache-related System Service calls enable the system to delete blocks and volumes from the cache, but these are not documented in the *GS/OS Reference* and can be used only by file system translators and other Apple-supplied system components.

`CACHE_ADD_BLK` ($01/FC08) is the System Service call for a routine that adds blocks to the cache. This call takes several parameters on GS/OS direct page (which is available to drivers and file system translators but not to applications), including information to identify the block by volume, device number, size and block number, and whether or not a write-deferral session is in progress. The Cache Manager is called, and it adds the block to the cache with no filtering. If necessary, the entire purgeable cache (the cache size minus all nonpurgeable write-deferred blocks) will be purged to add the block. The call will return with an error if a block could not be added to the cache, most likely because the entire purgeable cache was smaller than the block to be cached.

`CACHE_FIND_BLK` ($01/FC04) is the System Service call for a routine that finds blocks in the cache. It will search the cache for a specified block, returning a pointer to it if it is found. The cache can be searched by device number, so a device driver can find all blocks it has cached, or by volume ID, so that a file system translator can find all blocks it has cached (when a write-deferral session is in progress).

`MOVE_INFO` is the System Service call for a memory-moving routine. This routine is called by drivers and file system translators to move data in and out of the cache.

`SET_DISKSW` is the System Service call for a routine that kicks out all purgeable blocks in the cache belonging to a switched disk. If `SET_DISKSW` is called while a write-deferral session is in progress involving closed files on that device, GS/OS puts up a dialog box warning that the disk was prematurely ejected and that the disk's structure may be damaged.

There are two main parts of GS/OS that can use these System Service calls—drivers and file system translators.

The Role Played by Drivers

Drivers filter cache requests passed on from the application and file system translator levels. With every `Driver_Read` and `Driver_Write` command a driver gets from an application, it is passed instructions to take one of three possible caching actions:

- If the `cachePriority` word (on GS/OS direct page) is $0000, the block being read or written should not be cached. However, if the block is already in the cache, it must be updated on a write. Otherwise the next read of that block from the cache would return the wrong results.

- If `cachePriority` is nonzero with the high bit clear ($0001–$7FFF), the block should be cached as a normal, purgeable block.

- If `cachePriority` has the high bit set ($8000–$FFFF), the block should be cached as a deferred nonpurgeable block. This means a write-deferral session is in progress. In this case, which is only valid for `Driver_Write` calls (there are no read-deferral sessions in GS/OS), the driver should write the requested block only to the cache and not to the physical media. The end of the session will result in the driver's being called again to write all the cached blocks to the physical media.

There are instances, such as identifying volumes, in which a file system translator might wish to force a read from physical media rather than from the cache (if the block is in it). In these cases, the file system translator ID number on GS/OS direct page has the high bit set, telling drivers not to read the block from the cache.

A driver is not obligated to cache blocks when requested to, but instead can decide to disable caching completely or selectively. In some cases, the driver should refuse to cache *any* blocks. For example, this would be appropriate if the driver is for a device that cannot identify a disk-switched condition with any degree of reliability, as with a 5.25-inch disk. Since the driver can't call `SET_DISKSW` until it notices the disk has been switched—which could be well after the fact—a block in the cache for such a device might not be deleted when it should be, and thus the driver should refuse to cache any blocks in the first place. Or, for example, if a driver can actually read from the media faster than a block can be returned from the cache, it should refuse to cache blocks.

The Role Played by File System Translators

File system translators can initiate cache requests themselves, and can filter cache requests passed on from the application level.

When a file is opened and read or written, not all of the requests to a driver for information from the physical media are requests for data blocks from the file in question. Many of the read and write requests are for directories, file-system-specific data structures—such as key blocks and index blocks in ProDOS—and bitmap blocks. File system translators need these data structures repeatedly during file operations, and may ask that the blocks involved be cached. The ProDOS file system translator does this, caching all blocks it reads and writes that aren't passed on to the application. If the caching can be done, the file system translator gets much faster response time the next time it needs those blocks. On ProDOS disks, the caching of the volume directory and volume bitmap give tremendous speed increases since every file opening causes a read of the volume directory, and every write operation causes a read and write of the volume bitmap.

The file system translator may also cause caching of a different variety. When a write-deferral session is enabled, the file system translator changes the `cachePriority` field it passes to the drivers so that blocks are marked as write-deferred. This places them in the cache in a nonpurgeable state but not on the disk. The ProDOS file system translator asks that all of its ProDOS-specific directory, index, and bitmap blocks be placed in the cache write-deferred, but that all pure data blocks go straight out to disk without caching. This enables most devices to write data to contiguous areas of the disk, so that the head doesn't need to move back and forth writing directories and index blocks and updating bitmaps. All of that is done in one burst at the end of the session. Other file system translators may use the cache during sessions in ways that make sense for that particular file system. For example, the AppleShare® file system translator handles caching in a completely different way, not involving the GS/OS cache at all, since the media could change on the server from another workstation. Most file system translators, however, use the GS/OS cache in ways similar to the way the ProDOS file system translator uses it.

A file system translator can deny an application's caching request if it interferes with optimal system performance. For example, caching a ProDOS file being copied to a ProDOS disk during a session would slow things down, since cached file blocks would be continually removed from a full cache to make room for more write-deferred system blocks from the ProDOS file system translator. Write-deferral sessions are usually used to copy large numbers of files or create them from memory.

In either case, the files in question aren't likely to be read again, so the ProDOS file system translator eliminates possible overhead by denying requests to cache files during write-deferral sessions.

Resetting the Cache from the Application

Only one application-level call acts only on the cache—the `ResetCache` call (class one only, call number $2026). This call forces the cache to be reinitialized, purging all blocks that are in it and resizing if necessary. Do not issue this call while a write-deferral session is in progress; you can use `SessionStatus` to see if a write-deferral session is currently active.

If you're writing a utility program and suspect that a file system translator has cached something you don't really want around, calling `ResetCache` will ensure the cache is flushed. The cache is also resized from the battery RAM parameter. As discussed earlier, this parameter is currently $81 and represents the cache size in 32K increments, but this is not guaranteed. `ResetCache` is called by the RAM CDev to change the size of the cache.

To make the call, simply issue it with a parameter block pointer to a word of $0000. There are no parameters.

Where to Go from Here

This article has described the basics of the GS/OS cache, and has given you an idea of how an application requests caching and how such a request is fulfilled (or not fulfilled, if the request turns out not to be in the best interests of the system).

If you want to experiment with the effects of GS/OS calls that request caching, play around with these calls in the Exerciser that comes on the GSBug™disk. That's why it's there. It's come a long way from the ProDOS 16 Exerciser. It makes all the calls (in-line or stack-based, any class, with the exception of `ResetCache`), lists all devices, and catalogs directories to 255 levels. It also lets you choose any number of parameters for any class one call, except `ResetCache`, and allows you to modify memory through a built-in editor—you can visit the Monitor and return if you so choose. (Incidentally, the new calls for System Software 5.0 are coming in a revision soon.)

As always, help to Apple Partners on all matters, including GS/OS, is available on the AppleLink® network and MCI Mail from Apple II Developer Technical Support. If you're not an Apple Partner, you can often get help from knowledgeable programmers on third-party on-line services, usually in the "Developers" or "Development" forum.

Note: *develop* is published quarterly by Apple's Developer Press organization and is available by subscription for $30 a year from Kable-Publishers Aide, 425 N. Date Street, Suite C, Escondido, CA 92025.

Overview: Commercial Network Services

The following article is excerpted from "Make Mine On-line," The Apple IIGS Buyer's Guide, *Spring 1990.*

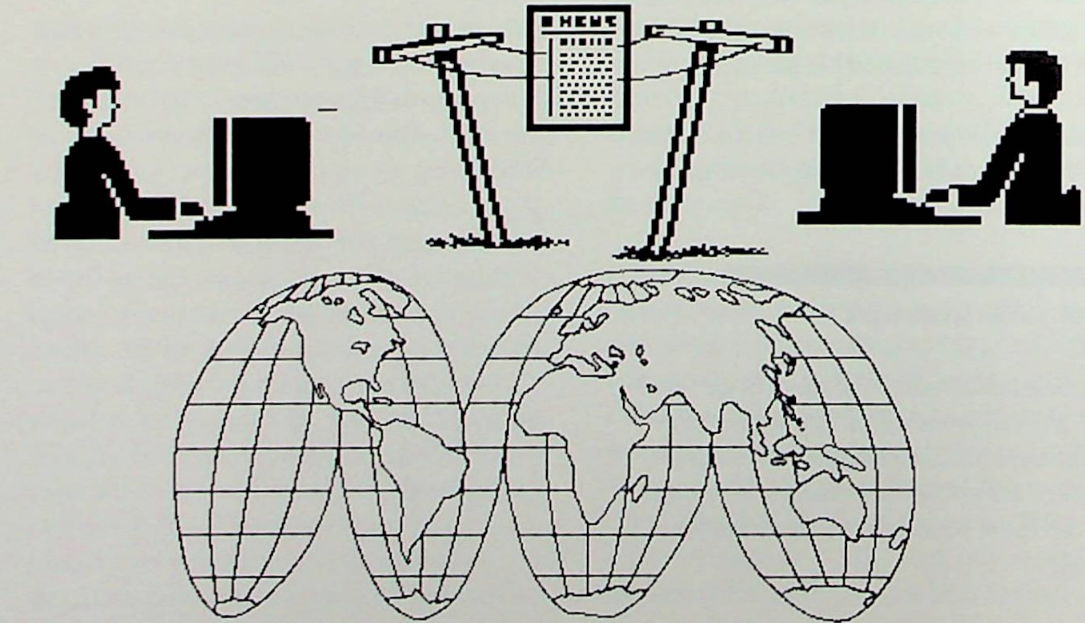
With an Apple II computer, a telephone, a modem, and communications software, you can join a community that's open to anyone who uses a computer. Commercial networks offer a wide variety of services, and all you need to access them is a terminal program configured for your hardware, and a modem with the appropriate cable for your computer.

America Online, CompuServe, and GEnie are the three networks that offer the most to the Apple II user. Although payment structures vary, they all charge you based on your "connect time"—the amount of time you spend using the system. To gain access, you must have a subscription and a password. Special offers are often packaged with new modems or telecommunications programs. Some offers waive the initial subscription fee; others provide a few hours free from connect charges. You can also contact the networks directly (see "For More Information," at the end of this article).

Each of the networks described in this article offers services for Apple II and IIGS users. They also provide diverse services such as on-line computer equipment purchase, airline schedules and reservations, news from the wire services, games you can play with other subscribers, and real-time chatting (an on-line "conference call," with all participants typing instead of talking).

America Online

America Online requires you to use specific, proprietary software. Provided free at sign-up time from Quantum Computer Services, the software uses pull-down menus, is mouse-driven, and can be used only with America Online.



Of major importance to Apple II users is the America Online Computing and Software area, where you'll find the forums and software libraries, Direct Connection (to third-party software and hardware developers), and the Magazine Rack (featuring on-line materials from major computer magazine publishers).

A forum is a special-interest group, and each forum maintains an area for messages, software, articles, and reviews. America Online forums for Apple II users include AppleWorks®, Art and Graphics, Communications, Utilities, Education, Games, Music, Productivity, and Word Processing applications.

Of the services described in this article, America Online is the least expensive, and its software libraries contain almost 6,000 public domain programs and files that you can download. Each forum offers a direct connection to hardware and software developers, including Activision, Applied Engineering, Beagle Bros, Berkeley Softworks, ByteWorks, Davidson and Associates, Electronic Arts, Infocom, Milliken, Orange Micro, PBI Software, Sierra On-line, and Spectrum Holobyte.

CompuServe

CompuServe Information Service, owned by H&R Block, is the oldest of the three networks and its index of services lists more than 550 areas.

The forums of interest to Apple II users are run by MAUG, the Micronetworked Apple User Group. Separate MAUG Apple II forums address Productivity, Fun, and Vendor Support. On-line technical support is provided by such companies as Activision, Applied Engineering, First Class Peripherals, Stone Edge Technologies, Broderbund, Beagle Bros, and Claris. Each MAUG forum offers a Message Base and a library. The Message Base is a discussion area where members can post and reply to questions. On a typical day, 100 to 200 messages may be posted, with one question generating many answers. Because CompuServe is so large, older messages must be deleted to keep within the MAUG forums' size limitation.

MAUG members have developed four different off-line software processors that allow you to automate all aspects of using the forums—resulting in faster access and lower connect charges. For example, Co-Pilot is a shareware program from Ken Glucksman that acts as a front end to either Talk Is Cheap or Point-To-Point Version 4.0. It's a GS/OS-specific program that uses the IIGS desktop environment. Co-Pilot lets you quickly access any forum, retrieve all messages, and log out. You can even prepare a list of software to be automatically downloaded from the libraries. You can then read and respond to messages off-line, without paying connect charges.

GEnie

Owned and operated by General Electric, GEnie has more than 100,000 members. Like CompuServe, GEnie has hundreds of general-interest forums and services, as well as several Apple II and IIGS special-interest groups, called Roundtables (RTs).

The two RTs of greatest interest to Apple II users are called A2 (general interest) and A2 Pro (programmer-oriented). On-line technical support is provided by companies such as Applied Engineering, Applied Ingenuity, Beagle Bros, CMS, Checkmate Technology, Chinook Technology, Zip Technology, Thirdware, A2-Central, Ohio Kache Systems, L&L Productions, Roger Wagner Publishing, SoftDisk, United Software Industries, First Class Peripherals, and Simple Software Systems International.

Of all the on-line networks, GEnie's software library collection is the largest, with more than 10,000 files and programs. The library menu is extensive and allows you to set custom options, such as displaying a list of all files in all categories, or a smaller list of all files added since your last network access.

Tom Hoover's GEnie Master, or GEM, shareware program is the A2 and A2 Pro off-line message system. GEM accomplishes the same service as that provided by CompuServe's Co-Pilot; it allows you to get in and out of the RTs as quickly as possible, thereby allowing you to save money. It works as a front end to Talk Is Cheap (from Carolina System Software). Reading messages and replying to them is done through AppleWorks.

Choosing a Network

America Online, CompuServe, and GEnie all offer valuable services for the Apple II user—each with a slightly different approach and different pricing structure. Note that all three networks either own or lease a long-distance telephone network, and hourly connect charges include telephone charges. The majority of users simply dial a local telephone number for access. If you live in a rural area, however, you may have to pay additional long-distance phone charges.

For More Information

America Online
Quantum Computer Services
8619 Westwood Center Drive
Vienna, VA 22182
1-800-227-6364

Miscellaneous costs:
Software and a manual are provided to all subscribers at no charge. The monthly $5.95 membership fee includes one hour per month free of connect charges.

Hourly connect charges:
$10 per hour for daytime usage; $5 per hour for evenings, weekends, and holidays.

CompuServe
CompuServe Information Service
P.O. Box 20212
Columbus, OH 43220
1-800-848-8199 or (614) 457-0802

Miscellaneous costs:
A membership kit retails for $39.95. The kit includes a manual and a $25 usage credit. After three months, a $1.50 per month service charge is initiated.

Hourly connect charges:
$12.50 per hour for 1200- or 2400-bps access; $6 per hour for 300-bps access. There is an additional $.30 per hour charge to cover telephone long-distance access charges.

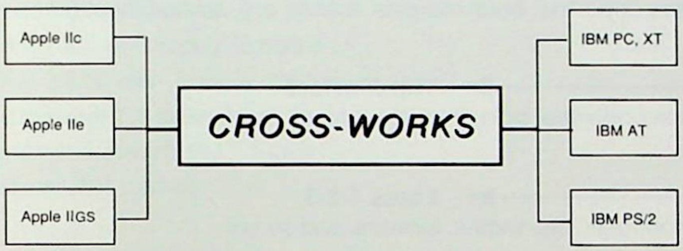
GEnie
General Electric Information Services
401 North Washington Street
Rockville, MD 20850
1-800-638-9636

Miscellaneous costs:
A membership kit retails for $29.95. The kit includes a manual and a $10 usage credit.

Hourly connect charges:
$18 per hour for daytime usage; $10 per hour for 2400-bps evening, weekend, and holiday access; $5 per hour for 300-bps evening, weekend, and holiday access.

Cross-Works 2.0: Apple II–to–IBM File Transfer

Cross-Works 2.0, from SoftSpoken, is a software application that provides Apple II users with a high degree of MS-DOS compatibility, while preserving their investment in the Apple II.



Cross-Works allows you to work on a document in AppleWorks and transfer it to WordPerfect on an MS-DOS machine—with its formatting intact. You can also transfer spreadsheets into Lotus 1-2-3 (with formulas as well as data), or databases into dBASE III, III Plus, or IV. All three functions transfer into Microsoft Works. In fact, you can transfer Apple II files into any MS-DOS-compatible application that reads a Lotus, dBASE, or Microsoft Works file format, a capability that provides almost complete file compatibility. The transfer works both ways; your Apple II can also receive files from the MS-DOS environment. (See the conversion chart on page 15.)

Cross-Works comes with an 8-foot universal cable that has all the necessary connections: a 25-pin male connector for an Apple II Super Serial Card (for an Apple IIe), a 5-pin connector for the Apple IIc, an 8-pin mini-DIN connector for the Apple IIGS or IIc Plus serial port, and 25-pin and 9-pin IBM-compatible serial connectors. After making the physical connection, you load the Cross-Works 2.0 program on each machine, select the files to be transferred, and initiate the transfer. Files travel across the cable at 19,200 baud and are fully converted into the requested format. For example, you might want to transfer a 30,000-byte AppleWorks spreadsheet file called FORECAST to an IBM AT. If you request the Lotus 1-2-3 format from the Cross-Works menu, you'll have an identical file called FORECAST.WK1 on the IBM disk in about 20 seconds.

If you already use an emulator, such as the Applied Engineering PC Transporter, you can convert files with Cross-Works and then transfer them with the emulator's transfer program.

If your computers are far apart, SoftSpoken offers an optional 50-foot cable. If it isn't practical to use a direct cable connection, you can use any Hayes-compatible modem, without additional communications software.

The Cross-Works program uses a standard AppleWorks interface and includes file utilities that allow you to rename, delete, or check that a file is identical on both systems. Preferences for default file conversions and screen colors can be saved; you don't have to reset them each time you use the program. Cross-Works warns you before it replaces existing files. It also supports translation of special foreign characters in word processing files.

The documentation explains limitations and offers suggestions for converting files. For example, when the source program has features that the target program lacks, the conversion may not be exact.

AUTOMATIC FILE CONVERSIONS

AppleWorks ↔ **Microsoft Works**
Word processor (keeps underline, bold, margins, and so on), spreadsheet (keeps formulas), and database

AppleWorks ↔ **WordPerfect**
Word processor (keeps underline, bold, margins, and so on)Version 4.1 through 5.1

AppleWorks ↔ **Lotus 1-2-3**
Spreadsheet (keeps formulas, cell widths, formats, and so on)

AppleWorks ↔ **dBASE III, III Plus, IV**
Database (also works with Q&A, Paradox, R:Base, FoxBASE, Clipper, and others

Automatic file conversion works with all versions of AppleWorks (1.0 through 3.0), except AppleWorks GS.

Cross-Works also converts:

- AppleWorks
  —to any generic word processor, text only,
  —to PFS First Choice (text only on word processor)
- ProDOS text file
  —to MS-DOS ASCII file
- Word Perfect 1.1 for Apple II
  —to WordPerfect 4.2 (preserves formatting)
- AppleWorks database
  —to Delimited ASCII database
  —to Lotus 1-2-3
- Any ProDOS file
  —to Any MS-DOS file (exact copy without translation)

To use Cross-Works 2.0, you'll need the following:

- An Apple IIe (with 128K of memory and a Super Serial Card), Apple IIc, Apple IIc Plus, or Apple IIGS personal computer

- An IBM PC, XT, AT, PS/2, or compatible with a serial port and DOS 2.0 or later

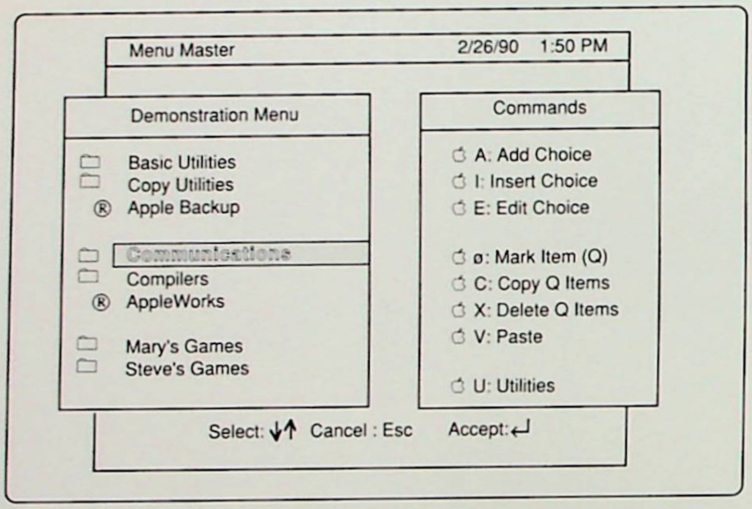
The package includes both 5.25-inch and 3.5-inch disks in both Apple II and IBM formats.

For further information, contact SoftSpoken as follows:

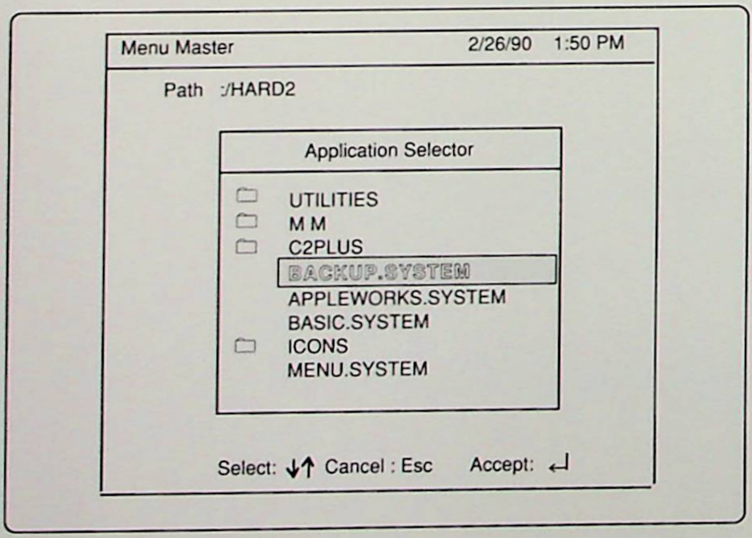
SoftSpoken, Inc.
P.O. Box 18343
Raleigh, NC 27619
(919) 870-5694
Fax: (919) 870-5696

Menu Master: Menu System for Launching Apple II Applications

Menu Master is a complete menu system for launching ProDOS, GS/OS, and DOS Version 3.3 applications. The program allows creation of customized menus and submenus, and provides automatic path finding, which simplifies the need to understand or manually set ProDOS paths. The Menu Master interface is easy to use; even a beginner can customize menus for launching applications. Simple keyboard commands allow selection of all features and functions.



Customized menus with simple keyboard commands



Automatic path finding

Users with any Apple II computer that has 128K of memory and an 80-column display can organize, launch, and return from ProDOS, GS/OS, and DOS 3.3 applications with Menu Master. The network version is compatible with all Apple II networks.

Menu Master provides the following additional advantages:

- It requires no memory expansion.
- It has password protection.
- It's mouse-compatible.
- It displays the time and date from any ProDOS clock.

For further information, contact Electronic Learning Systems, as follows:

Electronic Learning Systems, Inc.
2630 Northwest 39th Avenue
Gainesville, FL 32605
(904) 375-0558
1-800-443-7971
AppleLink address: D6034

ProDOS, GS/OS, and AppleTalk: Launching and Quitting

When using a shell program for the AppleTalk® network system, such as Apple's Aristotle™ software, you may notice that some applications quit to the shell from which they were launched, while others return directly to the File Server Log-On screen.

Here are some examples:

- You log on and go to Aristotle, the startup application. You launch the Finder from Aristotle and then launch "Application X" from the Finder. On quitting Application X, you are returned to the Finder, and on quitting the Finder (Special menu, Shut Down, Return to Launching Application), you are returned to Aristotle.

- You repeat this process to get to the Finder, but this time you launch "Application Y." On quitting Application Y, you are returned directly to Aristotle, even though you launched Application Y from the Finder.

In the two examples, behavior that occurs on launching and quitting is determined by the operating system for which the application is written: GS/OS or ProDOS 8.

The ProDOS 8 Quit command is a simple command; it doesn't remember the name of the application to return to. ProDOS applications quit to the program named Start in the GS/OS volume—in the examples above, the Finder. If the startup program has been set in the AppleShare Admin application, whatever program is set as the startup application is the application to which the ProDOS 8 programs return.

GS/OS applications remember the name of the program to return to. If properly written, a GS/OS application returns the user to the same application that launched the GS/OS program.

In the first example, with Aristotle (ProDOS 8) as the Admin set startup application, when you launch the Finder (GS/OS), then launch Application X (in this case, a GS/OS application), and quit Application X, it is correct to finish at the Finder. Quitting the Finder (GS/OS) then returns you to Aristotle, which originally launched the Finder.

In the second example, you launch Application Y (in this case, a ProDOS 8 application), then quit Application Y, and finish in Aristotle—correct behavior for the ProDOS 8 Application Y.

Because of an associated operating system issue, you might experience the following:

You put ProDOS applications stored on floppy disks into Aristotle menus. When you insert a disk and choose the menu item, the application launches properly and, on quitting, returns you to Aristotle. However, when you follow the same steps using Instant Pascal, the server connection is dropped somewhere, and quitting fails.

Instant Pascal is neither a ProDOS nor a GS/OS application; it has its own operating system with its own input/output routines—which know nothing about the network and disregard all network activities. Any program with an operating system that doesn't support AppleTalk cannot be used on the network.

✍

Apple II–to–IBM Host File Transfer

There are a couple of ways to transfer an ASCII file from an Apple IIe to an IBM mainframe without purchasing products specifically designed to do the job. The options are based on file transfer from the ASCII environment to the EBCDIC environment (microcomputer to mainframe host).

The first option is to send the ASCII file from the Apple IIe over a commercial network to a Macintosh or an MS-DOS-compatible PC connected to a mainframe. You can then make the ASCII-to-EBCDIC conversion and store the file on the mainframe. The transfer requires a mainframe-attached protocol converter that uses the XModem protocol, and an Apple IIe communications package that also supports XModem. Such a setup allows the Apple IIe to dial directly into the protocol converter and upload the file to the mainframe.

Another option is to dial the Macintosh or MS-DOS PC directly instead of paying for a commercial communications service. Because both computers require modems to access the commercial service, and because they probably use XModem for the transfer, you can connect them directly via communications software, using XModem for the transfer. Once transferred from the Apple IIe to the Macintosh or PC attached to the mainframe, the file can be uploaded from the Macintosh or PC using the host file transfer method.

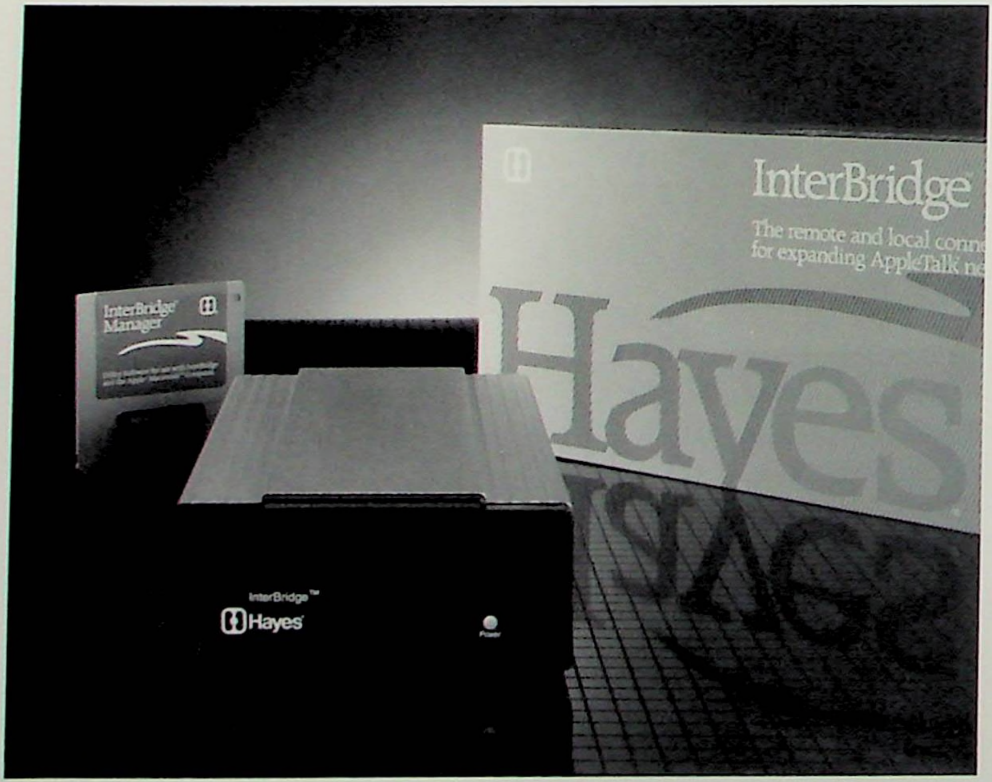
Hayes InterBridge: IIGS Remote Access to AppleShare

If you have an Apple II network and you want dial-in service from a remote Apple IIGS, one solution is to use the InterBridge, from Hayes Microcomputer Products, Inc. The InterBridge allows two networks, each equipped with an InterBridge, a Hayes Smartmodem, and a Macintosh computer, to access each other through either dial-up or leased telephone lines.

The Hayes InterBridge

The InterBridge product expands the connectivity of LocalTalk® networks by providing both local and remote bridge capabilities. Two InterBridges can connect two LocalTalk networks to enable a computer on one network to access computers, file servers, LaserWriter® printers, and other devices on the other network.

Equipping LocalTalk networks with InterBridges and modems allows access to similarly equipped remote networks for device sharing and data transfer. In addition, the remote bridge supports asynchronous or synchronous communication links over dial-up and leased lines.



Through a local or remote link, internetwork routing allows any node on an InterBridge network to address any other node on a similarly equipped network. Internetwork routing also permits multiple paths to a given destination.

InterBridge Manager software (for Macintosh computers) provides configuration and management options for ports, data transmission characteristics, zone name and network number alterations, and name selection for InterBridge units. The software indicates the availability of other InterBridge units and places calls for dial-up of remote bridges. Context-sensitive help screens are included, and a diagnostic window permits isolation of network or routing failures.

The InterBridge includes a complete hardware self-test at power-up with a blinking LED error indicator, and a nondisruptive diagnostics report that may be read at any time.

For further information, contact Hayes as follows:

Hayes Microcomputer Products, Inc.
P.O. Box 105203
Atlanta, GA 30348
(404) 449-8791

✍

Apple IIGS: Startup from an AppleShare Server Volume

To set up an Apple IIGS to start up from an AppleShare file server, follow these steps:

1. Using the Apple II Setup disk Version 2.1.1 (a Macintosh disk, provided with the Apple IIGS System Software), install the Apple II workstation software onto the AppleShare file server. Note that you must install Version 2.1.1 for use with GS/OS System Software 5.0.2 or later. Installation instructions are included in the *AppleShare File Server Administrator's Supplement for Apple II Workstations* and in the *AppleTalk Network User's Guide for the Apple IIGS*.

2. At the Apple IIGS, format a blank 3.5-inch floppy disk, and name it "AppleShare." Make sure your Control Panel slot settings are as follows:

   - For the 1-megabyte IIGS, Slot 1 should be "AppleTalk."
   - For the older IIGS, Slot 1 should be "Your Card," and Slot 7 should be "Built-in AppleTalk."

3. Using the Installer program on the GS/OS System Tools floppy disk, install the "AppleShare on 3.5" option onto the "AppleShare" floppy disk. After the installation is complete, restart the IIGS using the AppleShare disk.

4. From the Graphic Control Panel under the Apple menu on the IIGS desktop, mount the AppleShare server volume as the Administrator.

5. Run the Installer program again, and install "Server Network Startup" and "Server Quick Logoff" onto the AppleShare server volume. If you want to install more GS/OS drivers onto the server (for instance, if you need support for the Apple 5.25-inch disk drive or a printer), install them now. When installation is complete, quit the Installer.

6. From the IIGS desktop, open the AppleShare volume and highlight the System Folder, which contains part of the GS/OS operating system. Choose Icon Info from the Special menu, and set the access rights to Allow Everyone to See Files and See Folders, and click Apply to Enclosed Folders. Repeat the process for the Icons folder.

7. Set your IIGS to boot from the AppleShare volume by setting the Control Panel Startup Slot as follows:

   - For the 1-megabyte IIGS, the Startup Slot should be "AppleTalk."
   - For the older IIGS, the Startup Slot should be Slot 7.

8. From the AppleShare file server, set the users' startup application using the Admin program. If you want a user to start up to the GS/OS desktop, set the startup application to "Finder" (within the System Folder).

You can now start up the Apple IIGS from the AppleShare file server.

Apple IIGS Q&A

Q: I have an older Apple IIGS (with 256K of RAM on the logic board) and would like to upgrade to ROM Version 03. How can I do it?

A: The new Apple IIGS computer, which has 1 megabyte of RAM installed on the logic board, uses ROM Version 03, which differs significantly from the previous ROMs. Version 03 contains portions of the GS/OS operating system that were previously loaded during the startup process. Also, the physical logic board of the new IIGS is different; old ROMs don't work with the new board, and vice versa.

Because of the ROM changes that support the new logic board, it isn't possible to upgrade an older IIGS logic board to the Version 03 ROMs.

There are currently three different IIGS ROM versions:

- Version 03—Found in the new 1-megabyte IIGS. It displays "ROM Revision 03" at the bottom of the startup screen.
- Version 02—The current ROM for the older IIGS. It displays "ROM Revision 01" at the bottom of the startup screen.
- Version 01—The original Apple IIGS ROM. It doesn't display a revision code during startup. If you have this ROM version, you can upgrade your computer to Version 02 free of charge. Contact an authorized Apple service provider or your Apple sales representative.

Q: Is there a compatibility problem with the main logic board of the new 1-megabyte Apple IIGS and the DuoDisk® floppy disk drive? When I connect the DuoDisk to the disk port (SmartPort), the system will not start.

A: Yes. A compatibility problem with the DuoDisk and the new Apple IIGS main logic board requires a workaround: Instead of connecting the DuoDisk to the SmartPort, connect it to an interface card; code in the ROM doesn't handle the interface to a DuoDisk.

The following questions and answers are excerpted from "*develop, The Apple Technical Journal*".

Q. Why won't AppleTalk use the modem port on my ROM 01 IIGS?

A. If Slot 1 is set to "Your Card" in the Control Panel on a ROM 01 IIGS, AppleTalk will always use the printer port. AppleTalk will only use the modem port if Slot 1 is set to "Printer Port". This means that if you have a physical peripheral in slot 1 (such as a hard drive), AppleTalk will always use the printer port. ROM 3 IIGS machines require Slot 1 or Slot 2 to be set to "AppleTalk" to determine the network port, thus eliminating the confusion.

Q. What is the format of the IIGS Finder Data Files? There is no File Type Note describing the format.

A. The format of Finder data files is internal to the Finder. It is version-dependent and the information in the files can't be reliably used at this point, so the file format is not available.

Q. How can I control the speed of the Apple IIc Plus?

A. Firmware routines exist to allow the programming of the accelerator in the Apple IIc Plus. These routines are documented in the *Apple IIc Technical Reference, Second Edition,* available from APDA™(Apple Programmers and Developers Association).

Q. How can I use the "extra" keys on the Apple Extended Keyboard on my Apple IIGS? What value do they return?

A. The following chart shows what the "extra" keys and other extended keys of the Apple Extended Keyboard return on the Apple IIGS. All of the keys turn on bit 4 of the Modifier Key register at $C02 (see page 124 of the *Apple IIGS Hardware Reference*) to distinguish these keys from the "regular" keys.

Key	ASCII character	Key	ASCII character
F1	z	F10	m
F2	x	F11	g
F3	c	F12	o
F4	v	F13	i
F5	` ($E0)	F14	k
F6	a	F15	q
F7	b	help	r
F8	d	home	s
F9	e	page up	t
		page down	u
		end	w
		del |X>	y

Apple II: Using Macintosh SCSI Drives

The following article is excerpted from "Apple Clinic," inCider *magazine, March 1990.*

SCSI-compatible hard disk drives marketed for the Macintosh computer can also be used with Apple II computers. Most external hard disk drives can operate from the Apple II SCSI Card (Revision C).

Although a drive may be formatted for the Macintosh, making it impossible to start up from an Apple II, it is generally necessary only to reformat the hard disk with an Apple II System Disk and copy ProDOS and your startup application to it.

Apple IIGS users must first create a startup system disk that includes the SCSI hard disk driver. (Use the Installer program on your GS/OS System Tools disk, and reset the Control Panel assignment for the slot where you've installed the SCSI card to "Your Card.")

After starting up from the new system disk, use the Advanced Disk Utility on the System Tools disk to format and partition the SCSI hard disk. Use the Installer program to copy the GS/OS system files, the SCSI hard disk driver, and any other system files (such as printer drivers) to the first partition of your hard disk drive.

✍

How to Tell an Apple IIe from an Apple IIe Enhanced

When the Apple IIe was introduced as the Apple IIe Enhanced, the CPU was changed from a 6502 to a 65C02, and the ROM was upgraded.

The following information will help you to determine whether a IIe computer is enhanced:

- An easy way to check for enhancement is look at the startup screen display. An unenhanced computer displays "Apple][," whereas the enhanced model displays "Apple //e." Another check is to enter commands in lowercase; if they are accepted in 40-column mode, the enhanced ROMs are installed.

- The 6502/65C02/65SC02 CPU chips usually carry a 6502 family number. Some chip number designations have additional letter and number prefix and suffix extensions. The Apple part is labeled with the Apple part number 338-6503. (Note that it is possible to buy and install a 65C02 chip in an Apple IIe without the ROM upgrade; the chips are available from many sources. However, you wouldn't have the enhanced features, because firmware support is required via the ROM upgrade kit.)

- You'll be able to access the Mini-Assembler on an enhanced Apple IIe. Enter the Monitor by performing a call to location -151; that is, from the Applesoft prompt, type the following:

  ```
  CALL -151 [Return]
  ```

  An asterisk prompt appears on the left side of the screen. Type an exclamation mark to enter the Mini-Assembler. If the Mini-Assembler is present, an exclamation mark replaces the asterisk prompt on the left side of the screen. Pressing Control-Reset returns you to Applesoft.

- In an enhanced Apple IIe computer, MouseText characters replace the redundant set of uppercase inverse characters in the character generator ROM of the IIe (marked as the video ROM on the logic board). Some older programs used the redundant uppercase inverse characters; some new programs use the MouseText characters. If you run a program that uses the inverse uppercase characters on an enhanced IIe, the characters will display as MouseText. Similarly, if you run a program that uses MouseText characters on an unenhanced Apple IIe, the inverse characters will display.

- All platinum Apple IIe computers have the enhancements described in this article.

Apple IIe/IIe Enhanced: Distinguishing Characteristics

	Apple IIe	Apple IIe Enhanced
Screen display at power on	Apple][	Apple //e
Accepts lowercase commands in 40-column mode	No	Yes
Mini-Assembler available	No	Yes
MouseText characters present in video ROM	No	Yes
CPU designation	6502	65C02 or 65SC02
ROM at location EF	PN 342-0134	PN 342-0303
ROM at location CD	PN 342-0135	PN 342-0304
Video ROM	PN 342-0133	PN 342-0265

✍

Aristotle: How to Run an Applesoft BASIC Program

To run an Applesoft BASIC program from Aristotle, follow these steps:

1. On the server, create a folder that contains the Applesoft program and BASIC.System.

2. Rename the Applesoft program "STARTUP."

3. In the Aristotle Management program, add the BASIC program to the class list, using any appropriate name. Set the path as follows:

   ```
   /{servername}/{foldername}/{any  other  foldernames}/BASIC.SYSTEM
   ```

4. Set the prefix to the Applesoft program pathname. BASIC.System looks for a program with the name "STARTUP" in the current folder (subdirectory) and launches it.

When you're finished with the Applesoft program, the BYE command doesn't return you to Aristotle. Because BASIC.System was the launching program, the BYE command attempts to run BASIC.System. To get back to the Artistotle menu, you must launch Aristotle again—which requires some programming.

The example code that follows assumes that the Aristotle folder is located at the server volume root level with a pathname in this format:

```
/{servername}/ARISTOTLE/MENU.D/DISPLAY
```

The code finds the servername and the places where `{servername}`appears in the pathname:

```
9900   D$=CHR$(4)
9910   PRINT  D$;"PREFIX"
9920   INPUT  PR$
9930   FOR I = 1 TO LEN(PR$)
9940   IF MID$ (PR$, 1+I,1) = "/" THEN 9990
9950   RT$=RT$+MID$(PR$1+I,1)
9960   NEXT I
9990   PRINT  D$;"-/";RT$;"/ARISTOTLE/MENU.D/DISPLAY”
```

Here's an explanation of what takes place:

9900 – Set D$ to Control-D.
9910 – Request the current PREFIX.
9920 – Put the current PREFIX into PR$.
9930 – Start loop to extract the servername from the PREFIX.

9940 – Skip the first character of PREFIX (/) and look for the second slash. When the second slash is found, jump to 9990.

9950 – Put all the characters up to the second slash into the variable RT$.

9960 – Increment, and do again until done.

9990 – Run the Aristotle menu display program.

If you use such code as a stand-alone program, place it in the folder with the Applesoft program. After quitting the Applesoft program launched from Artistotle, type the following:

```
RUN {name of program}
```

If you add the code to the Applesoft program, make sure that its lines are the very last ones executed by the BASIC program. (Remember that the line numbers need to be appropriate for the program to which you add them.)

✍

Aristotle and RAM

If you set 256K minimum RAM on the RAM disk in an Apple IIGS, or 256K of RAM on the expansion card in the Apple IIe, Aristotle allows you to manage up to 150 applications on the server and use them to create up to 50 menus for up to 500 users. (Aristotle calls the menus "classes.")

You can raise these limits by increasing the amount of RAM. With each 256K increment—to a maximum of 1 megabyte—you can manage 150 more applications, 50 more menus, and 500 more students.

If you're using an Apple IIGS, however, remember that 256K of the RAM on the expansion card is needed for the workstation to access the file server. In fact, more memory may be needed if the user has desk accessories or other memory-resident programs.

✍

Using a Switchbox with the ImageWriter II

If you have been unable to print from multiple systems by using a switchbox with an ImageWriter® II printer, you can solve the problem by using XON/XOFF handshaking.

This article explains what happens in a setup that includes an ImageWriter II, an Apple IIGS, an Apple IIe with a Super Serial Card, and two commercial switchboxes—one make-before-break, and one break-before-make. Line status is monitored with a data specification RS-232 break-out box, and a Fluke 8050A (or similar) digital multimeter is used to measure voltage levels.

Transitions on the transmit line cause DTR (Data Transfer Ready) from the printer to go into the high state when you switch between the system. This effectively shuts off communications with the connected computer. If the printer is reset before switching to the other system, it prints correctly. If there is no reset when switching, you must reset the printer either by using the select button or by turning off the printer and turning it on again. Either of these actions resets the DTR line and re-establishes communications with the computer.

If you use a switch box with a make-before-break switch, the switch doesn't work between the Apple IIe and Apple IIGS systems because of the difference in system voltage levels. The Apple IIGS system output device, AMD 26LS30, is supplied with +/-5VDC; the Super Serial Card output device, TI 75189, is provided with +/-12VDC. This means that the Apple IIGS outputs -4.7 volts for a "Low" signal, whereas the Apple IIe outputs -9.7 volts.

With a break-before-make switch, when the switch opens between contacts the voltage level rises to Ground, then drops to a Low level again. The positive transition to Ground causes the printer to see either a start bit or a DSR-level shift from the computer, setting then sets DTR false. Current could flow from the Apple IIe into the Apple IIGS output because of the different voltage level in the output "Low" signal, causing damage to the interface chips in the Apple IIGS.

The Workaround

A better approach involves using XON/XOFF handshaking. The printer sends/receives the XON/XOFF handshake over the normal data communications lines—pins 2 and 3—and although these lines are affected by the level transitions caused by the switch action, the transitions are not recognized as XON/XOFF by the printer or the computer. Because the equipment doesn't see the transitions as requests to halt data flow, there is no attendant response to incorrect data by the printer, and the information flow continues.

Change to XON/XOFF handshaking on the ImageWriter II by setting SW 2-3 open. The problem of not being able to print from multiple systems should vanish.

To use the XON/XOFF handshake with older software, use a preboot program to set up the Super Serial Card before using the software. On the Apple IIGS, if you cannot gain access to the Control Panel from within the application, you must also use the preboot.

The preboot program should look like this in Applesoft, using either DOS 3.3 or ProDOS:

```
10  D$=CHR$(4)
20 PRINT  D$;"PR#1"
30 PRINT  CHR$(9);"X  E";CHR$(13)
40 PRINT  D$;"PR#0"
```

Enter the program into the system in immediate mode before starting up, or save it on disk as either a HELLO (DOS 3.3) or a STARTUP (ProDOS) program to run before the software boots.

The Explanation

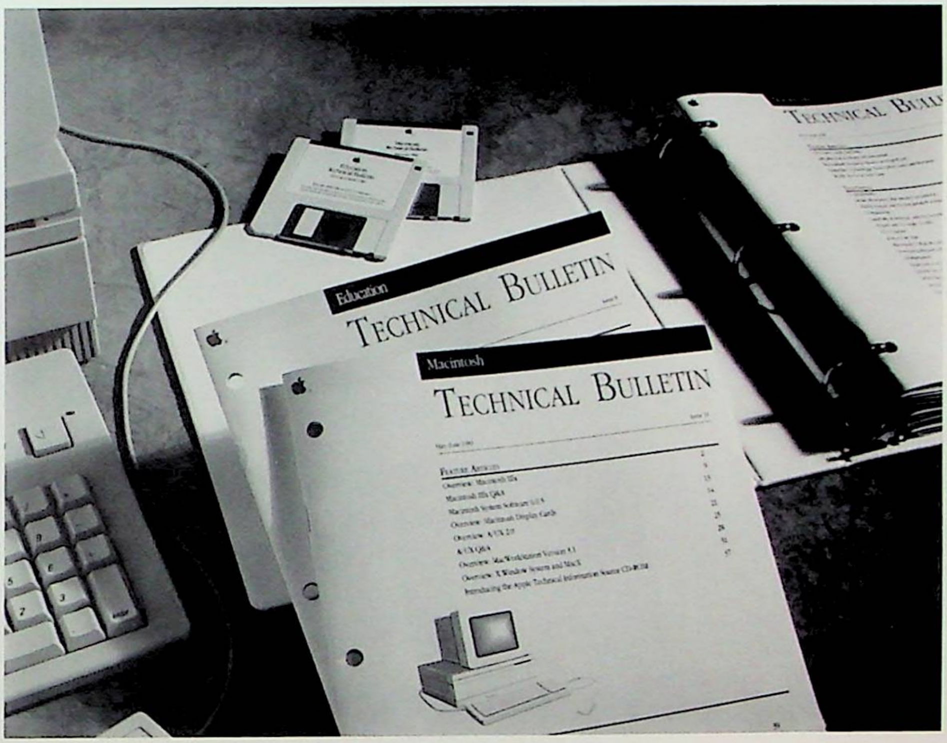
ImageWriter II printers use 26LS32 chips for receivers. Designed for high-speed use, these chips are very sensitive to level transitions. By contrast, the ImageWriter (I) printer uses 75189 receiver chips, which are functionally equivalent to the 1489 line receiver. These chips are less sensitive to level transitions because they are designed to operate at lower frequency limits.

✍

Apple Announces the *Education Technical Bulletin*

Beginning with the October–November 1990 issue, the *Apple II Technical Bulletin* becomes the *Education Technical Bulletin*—a change designed to better meet the information and support needs of its audience, virtually all of whom are Apple K–12 education customers.

The *Education Technical Bulletin* will cover both Apple II and Macintosh solutions for the K–12 marketplace, maintaining its focus on Apple II products and solutions, and adding coverage of the Macintosh solutions now widely used in the K–12 environment. The *Apple II Technical Bulletin* will no longer be available.



Stack of Back Issues

Apple is also enhancing the Technical Bulletin product line: Current and new subscribers will receive a HyperCard® stack containing all the back issues of the version of the Technical Bulletin to which they subscribe, starting with the first issue of the publication in 1988.

Current subscribers will receive a stack of the articles from 1988 and 1989 in the September–October *Macintosh Technical Bulletin* or the October–November *Education Technical Bulletin;* new subscribers will receive the disk as part of the Starter Kit. The complete set of articles from 1990 will be included with the January–February *Macintosh Technical Bulletin* and the February–March *EducationTechnical Bulletin.*

For more information about the *Education Technical Bulletin,*
please contact your Apple representative or
authorized Apple reseller.

Education Technical Bulletin
Part No. B0342LL/A
Suggested Retail Price: $129/per year

Important Information about the Apple II High-Speed SCSI Card

To use the Apple II High-Speed SCSI Card with an Apple IIGS and System Software Version 5.0.2, you must have at least 768K of RAM; 1 megabyte of RAM is recommended. Also, please note that the Apple II High-Speed SCSI Card will not work with the SCSI driver that comes with GS/OS; you must use the driver that comes with the card.

Install the updated SCSI driver using the Installer program on the utility disk that comes with the new SCSI card. Both the driver and the Installer program are in the "APPLEIIGS.FILES" folder.

Here are the steps to correctly install the new SCSI driver on an unformatted hard disk:

1. Start up the Apple IIGS with a copy of the GS/OS Version 5.0.2 System Disk.
2. Run the Installer program from the APPLEIIGS.FILES folder on the High-Speed SCSI Card utility disk, installing the SCSI Hard Disk option onto the copy of the System Disk.
3. Quit the Installer and shut down the system.
4. Connect the High-Speed SCSI Card and the hard disk drive to the Apple IIGS, and turn on the hard disk.
5. Start up the IIGS with the copy of the System Disk. The hard disk drive icon appears.
6. Select the hard disk icon and choose Erase Disk from the Special menu. Simply "OK" any questions about interleave factors. (Note that if the IIGS does not immediately recognize the hard disk, it will automatically ask you to initialize it.)
7. Install GS/OS onto the hard disk.
8. Install the new SCSI drivers onto the hard disk.

If the Apple II High-Speed SCSI Card is replacing an older SCSI interface, and the hard disk contains information, follow the same procedure but omit steps 6 and 7.

Apple II Technical Bulletin

Managing Editor: Armi Costello
Technical Editor: Shelley Bowers
Writer: Jennifer Woodul
Editor: Teri Thomas
Copy Editor: Van Goode
Production Manager: Susan Moore
Contributor: Matt Deatherage, John Love, and Daniel Camper

Special thanks to Apple's Technical Communication group and Technical Information Services group for their written contributions and technical expertise.

If you have any comments or suggestions, please contact:

Technical Bulletins
Apple Computer, Inc.
900 East Hamilton Avenue, M/S 72L
Campbell, CA 95008
AppleLink: TECHBULLETIN

The Apple II Technical Bulletin provides users with current technical information about Apple products. Featured topics may be under consideration, and new product versions may be under development. Therefore, it's appropriate to consider this publication as a bulletin that is correct at the date of publication, rather than as a long-term reference source.

Technical Bulletin Subscriptions

Apple Technical Bulletins are bimonthly publications designed to provide timely and useful technical information to Apple II and Macintosh users, particularly those who provide technical support to other users. Articles about new products, compatibility and connectivity issues, "power user" features, and product updates and upgrades help you use your Apple hardware and software more effectively and efficiently. Two editions are available, for an annual subscription price of $129 each: the *Education Technical Bulletin* (order number B0342LL/A) and the *Macintosh Technical Bulletin* (order number B0133LL/A). Each edition is published six times a year. Subscribers also receive the full text of each issue on disk. A storage binder is included.

Apple Technical Bulletins are also available as part of the Apple Software Update Program. For more information, contact your authorized Apple reseller or Apple sales representative.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576